7/7/03

REMARKS

Official

Claims 47-80 are pending. The Examiner allowed claims 47-58 and 74.

Claims 63-67 were objected to as dependent upon a rejected base claim. Therefore, claim 63 has been amended to incorporate all of the limitations of base claims 60-62. Claims 64-67 depend from claim 63. Since these claims were already allowable, the amendment is one of form, but not content.

Claim 80 has been added. Claim 80 depends from claim 58. Since claim 58 was allowable, claim 80 should also be allowable. No new matter has been added.

Information Disclosure Statement

An IDS was filed on July 3, 2003 by mail. A courtesy copy of the PTO Form 1449 is enclosed. Consideration of this IDS is respectfully requested.

The Director of Patents initiated reexamination of the parent application of this Application. The reexam control number is 90/006,529. A first Office action has not been mailed in the reexam.

Claim Rejections - 35 USC § 103

The Examiner rejected claims 60-62, 68-73, and 76-79 under 35 USC § 103(a) as obvious from Hsu (USP 5,584,023) in view of Brundrett et al (USP 6,249,866). These rejections are respectfully traversed. Claims 60, 68, 69, 71, 76 and 77 are independent.

Hsu is directed to a computer system including a transparent and secure file transfer mechanism. In Hsu's computer system, all encryption and decryption is performed using a single algorithm.

Brundrett is directed to a file system which includes transparent file encryption and decryption capabilities. Brundrett teaches that the user can choose among available encryption algorithms.

Claims 60 and 76-78: The Examiner still has not made a *prima facie* showing of obviousness of claims 60, 76-78, and these claims are not obvious from Hsu in view of Brundrett.

Claim 76 includes the step of:

> generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file.

Claims 77 and 78 includes the step of:

> testing the encrypted data identifier after decryption by regenerating the encrypted data identifier and ascertaining that they are the same.

Claim 60 includes both of these steps.

The Examiner contends:

> Hsu does teach or suggest appending an enode data structure (file identifier) to a regular file after the file has been transformed through the use of an encryption table which serves as the encryption key. The encryption table (encryption key) is formed through a shuffling / index value substitution function applied to the password key and seed table (see column 12, line 25 to column 12, line 26). Through this process data values (data identifiers) are created and are associated arithmetically to the decryption index values of the decryption table (see column 12, lines 27-49). The contents of the identified enode structure (file identifier) can be used in the authentication of the encrypted data. This file identifier has associated data identifiers which are part of the encryption table (encryption key).

This does not support the rejection. Hsu merely validates the accuracy of the encryption and decryption of the encryption password key, not the file itself [See column 14, lines 53-58]. Further, the data values in Hsu are generated only one time, prior to encryption and are stored both in the encryption table and decryption table [See column 14, lines 18-22, lines 43-48, and lines 53-58]. Although authentication is performed in Hsu, the method lacks the step of "regenerating" a "data

identifier" based on the post-decryption file. Therefore, Hsu fails to teach or disclose the method of regenerating a data identifier based on the post-decryption file.

Moreover, the data values are not generated based on the file, but are created by the "shuffle function" based on the user entered "password key" and a "predefined seed table" [See Column 11, line 45 – Column 12, line 49]. Therefore Hsu also fails to teach or disclose the step of "generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file". Hence, Claims 60, 76, and 77 are non-obvious in light of Hsu.

Yet, the Examiner contends:

> When Brundrett verifies the integrity and source of the encryption key, the data structure which holds data identifiers, is compared to verify the structure (see column 16, lines 5-16).

This does not support the rejection. Brundrett is merely validating the session key encryption / decryption of the File Encryption Key which is passed between the EFS driver (Figure 46) and the EFS service (Figure 50) [See Column 4, line 54 – Column 5, line 9 and Column 15, line 51 – Column 16, line 16].

In fact, there are three primary encryption / decryption schemes within Brundrett. One encryption / decryption scheme uses the File Encryption Key (FEK) to encrypt / decrypt the user's file [See Column 9, line 66 – Column 10 line 1]. The second encryption scheme uses the Public Key of the user to encrypt the File Encryption Key and the Private Key of the user to decrypt the FEK [See Column 10, lines 10-14]. The third encryption scheme is where the EFS driver (Figure 46) and EFS service (Figure 50) need to communicate. In the third scheme, a symmetric Session Key (SK) is utilized to encrypt data that is communicated between the EFS driver (Figure 46) and the EFS service (Figure 50) [See Column 4, Lines 54-61].

Within Brundrett, when a file is initially created and is designated to be saved in a pre-designated encrypted directory of a non-volatile disk, the FSCTL_SET_ENCRYPTION command is

issued in order to turn on the encryption bit for a stream [See Column 16, lines 5-9]. The FSCTL_SET_ENCRYPTION command is accompanied by a data structure (Figure 8) containing:

> A public code so that NTFS 28 can differentiate between the two types of FSCTL calls;
>
> An EFS subcode to more particularly define the operation for the EFS driver 46 and/or the EFS service 50;
>
> EFS data containing the FEK;
>
> The FEK encrypted with the Session Key;
>
> Optionally EFS metadata.

[See Column 16, Lines 9-11 and Column 15, line 60 – Column 16, line 1].

Except for the public code, the information within the data structure is encrypted with the session key. [See Column 16, line 1 – Column 16, line 4]. Within the data structure, "The FEK encrypted with the Session Key" is also encrypted. (In simple terms, the FEK is listed in the data structure both in raw form and session key encrypted form. Then the data structure, as a whole, is encrypted with the session key.)

In Brundrett, for purposes of verifying the integrity of the data structure (note that the EFS driver (Figure 46) and the EFS service (Figure 50) pass both the FSCTL_SET_ENCRYPTION command and the data structure, containing the FEK, between each other), the data structure is decrypted utilizing the symmetric system key. After the data structure has been decrypted, it contains both a FEK and a session encrypted FEK. The session encrypted FEK is then decrypted utilizing the symmetric system key. The verification step is accomplished by comparing the FEK values described in this paragraph. If they match, then the communication between the EFS driver (Figure 46) and the EFS service (Figure 50) was not compromised during the session key encryption and decryption process [See Column 16, line 12 – Column 16, line 16].

Nothing in Brundrett regarding the verification of the session key encryption and decryption of the File Encryption Key teaches or suggests a method including the step of "regenerating" a "data

identifier" based on the post-decryption file. Even if a person skilled in the art of transparent encryption and decryption methods combined Hsu and Brundrett, nothing is disclosed in either and nothing is taught in either to "regenerate" a "data identifier" based on the post-decrypted file for purposes of validation. Therefore, Claims 60, 76, and 77 are not obvious from Hsu in view of Brundrett and should be allowed to issue.

Claim 61: The Examiner still has not made a *prima facie* showing of obviousness. Claim 61 depends on claim 60 and recites an additional step:

> selecting the file from within the contents of a second file that is larger than the file.

The Examiner has not shown how Hsu or Brundrett disclose, teach or suggest this step. In fact, this step is not disclosed, taught or suggested by Hsu or Brundrett.

The Examiner contends:

> Hsu does teach or suggest files are referenceable (selected) via directories (second larger files) which are themselves standard files maintained on a disk (see column 6, lines 44-51).

This does not support the rejection. The directory in Hsu does not have as its "contents" another file. In Hsu, "directories are maintained on disk as standard files containing specifically structured data." However, the "directory file data" includes only a "pointer to a disk based structure of disk inode entries" and does not contain the file [See Column 6, lines 44-53]. Hsu's directory merely stores references to where the files are located. Moreover, Hsu does not disclose, teach or suggest "a second file that is larger than the file." In fact, Hsu has no disclosure as to relative file sizes. Therefore, claim 61 is not obvious from Hsu in view of Brundrett and should be allowed to issue.

Claim 62: The Examiner still has not made a *prima facie* showing of obviousness. Claim 62 depends on claim 61 and recites that "the encrypted file is placed in a container." The Examiner has not shown how Hsu or Brundrett disclose, teach or suggest this step. In fact, this step is not

disclosed, taught or suggested by Hsu or Brundrett. The Examiner contends:

> Hsu does teach or suggest the file is placed in a container (inode entries) which store specifically relevant information describing the file (See column 6, lines 51-57).

This does not support the rejection. Hsu's "inode entries" merely are descriptive data strings "describing, among other things, the protection mode, owner, user group and size of a particular data file" [See Column 6, lines 53 – 56]. An inode entry does not contain the encrypted file. Since nothing in Hsu or Brundrett either on their own or in combination suggests or teaches that "the encrypted file is placed in a container", claim 62 is not obvious from Hsu in view of Brundrett and should be allowed to issue.

Claims 68, 69, 71 and 79: Examiner has not made a *prima facie* showing of obviousness of claims 68, 69, 71 and 79. These claims include the steps:

> inputting a decryption key with a decryption key value;
>
> validating the decryption key value with the key value associated with the file identifier.

In fact, these steps are not disclosed, taught or suggested by Hsu or Brundrett. The Examiner contends:

> Hsu does teach or suggest the above along with teaching the through the transformation, data values (data identifiers) are created and are later used to generate the decryption index values (decryption key) needed in the validation process (see column 14, lines 50-67).

This does not support the rejection. Hsu's validation process compares a text string that resides both in the enode and the in-core inode [See Column 14, lines 50-54]. Hsu's encryption method requires that the user enter a password key. This password key and a random mapping seed table are processed through a shuffle function that generates both the encryption table and the decryption table [See Column 11, line 45 – Column 12, line 29]. The password key is encrypted

using the encryption table and is then appended to both the file's enode and the in-core inode [See Column 12, line 50-53, Column 14, lines 42 – Column 15, line 4 and Column 17, lines 4-13].

Hsu's validation process includes decrypting the enode using the decryption key and then comparing the text string located in the "magic_text" field of the decrypted enode to the corresponding text string located in the in-core inode [See Column 13, lines 56-65 and Column 14, lines 53 – Column 15 line 4].

Unlike claims 68, 69, 71 and 79, Hsu's decryption key used to decrypt the enode is not compared to anything during the validation process. And since Hsu's decryption key is not compared to anything associated with the file identifier or enode, Hsu fails to teach or suggest:

> inputting a decryption key with a decryption key value;
>
> validating the decryption key value with the key value associated with the file identifier.

The Examiner cited nothing in Brundrett that either on its own or in combination with Hsu would teach the above steps recited in Claims 68, 69, 71 and 79. Therefore, Claims 68, 69, 71 and 79 are not obvious from Hsu in view of Brundrett and should be allowed to issue.

Claim 70: Claim 70 is dependent on and contains all the limitations of claim 69. Since Claim 69 is not obvious from Hsu in view of Brundrett, claim 70 is also non-obvious from Hsu in view of Brundrett and should be allowed to issue.

Claims 72-73: Claims 72-73 are dependent on and contain all the limitations of claim 71. Since claim 71 is not obvious from Hsu in view of Brundrett, claims 72-73 are also not obvious from Hsu in view of Brundrett and should be allowed to issue.

Claim Rejections - 35 USC § 103

The Examiner rejected claims 59 and 75 under 35 USC § 103(a) as obvious from Hsu in view of Brundrett and Finley (USP 5,815,571). These rejections are respectfully traversed. Claims 59 and 75 are independent.

Finley is directed to a computer system with secure data paths and a method of protection. Finley's object is to prevent harm from viruses, essentially by quarantining all incoming data before allowing the data to be moved to a normal workspace. Finley teaches that, instead of a single computer, there should be three computers: a main computer, a security computer and a test computer. Finley refers to the security computer as a "firewall." All security functions are implemented in the security computer, which "must not" execute user programs. Before a new program which has been downloaded from the Internet can be run on the main computer, it is first run on the test computer.

Claim 59: Claim 59 is allowable for the same reasons as claims 68, 69, 71 and 79. Claim 59 includes the steps:

> inputting a decryption key with a decryption key value;
>
> validating the decryption key value with the key value associated with the file identifier.

Since Finley does not teach or suggest the above listed steps and nothing in Finley, when used in conjunction with Hsu and Brundrett would teach or suggest the above listed steps, claim 59 is not obvious from Hsu in view of Brundrett and Finley. Therefore claim 59 should be allowed to issue.

Claim 75: Claim 75 is allowable for the same reasons as claim 71. Claim 75 contains each and every step and limitation of claim 71. Claim 75 simply adds three additional steps:

> running a virus scan program on the file before it is encrypted;
>
> validating a decryption key value with the key value associated with the file identifier;
>
> using the key value and the algorithm to decrypt the file.

Since Finley does not teach or suggest the above listed steps and nothing in Finley, when used in conjunction with Hsu and Brundrett would teach or suggest the above listed steps, claim 75

Official 7/7/03

is not obvious from Hsu in view of Brundrett and Finley. Therefore claim 59 should be allowed to issue.

Conclusion

It is submitted, however, that the independent and dependant claims include other significant and substantial recitations which are not disclosed in the cited references. Thus, the claims are also patentable for additional reasons. However, for economy the additional grounds for patentability are not set forth here.

In view of all of the above, it is respectfully submitted that the present application is now in condition for allowance. Reconsideration and reexamination are respectfully requested and allowance at an early date is solicited.

The Examiner is invited to call the undersigned attorney to answer any questions or to discuss steps necessary for placing the application in condition for allowance.

Respectfully submitted,

Date: July 7, 2003

Steven C. Sereboff, Reg. No. 37,035

Joel G. Landau, Reg. No. P-54,732

SoCal IP Law Group
310 N. Westlake Blvd., Suite 120
Westlake Village, CA 91362
Telephone: 805/230-1350
Facsimile: 805/230-1355
email: info@socalip.com

PTO/SB/97 (12-97)
Approved for use through 9/30/00. OMB 0651-0031
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Attorney docket no.: M000-P02003US

Certificate of Transmission under 37 CFR 1.8

Name (office): M. Smithers (2134)

PTO facsimile number: 703/746-7239

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office

on 07-Jul-2003
Date

Signature

Steven C. Sereboff
Typed or printed name of person signing Certificate

Note: Each paper must have its own certificate of transmission, or this certificate must identify each submitted paper.

Enclosures:

Response to Office Action, 22 pages

Copy of Information Disclosure Statement, 6 pages

Fee Transmittal Form, 1 page

Credit Card Payment Form, 1 page

TOTAL including this cover sheet: 31 pages